



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,829	05/24/2001	Paul V. Werme	83019	2366

23501 7590 06/20/2006

NAVAL SURFACE WARFARE CENTER, DAHLGREN DIVISION  
OFFICE OF COUNSEL, CODE XDC1  
17320 DAHLGREN ROAD  
DAHLGREN, VA 22448-5110

EXAMINER
----------

KISS, ERIC B

ART UNIT	PAPER NUMBER
----------	--------------

2192

DATE MAILED: 06/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/864,829

Applicant(s)

WERME ET AL.

Examiner

Eric B. Kiss

Art Unit

2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 06 March 2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 38-72 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 38-42, 44-64 and 66-72 is/are rejected.
- 7) ☒ Claim(s) 43 and 65 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

## **DETAILED ACTION**

### ***Continued Examination Under 37 CFR 1.114***

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on March 6, 2006 has been entered.

Claims 38-72 are pending.

### ***Response to Arguments***

2. Applicant's arguments with respect to claims 38-72 have been considered but are moot in view of the new ground(s) of rejection.

### ***Claim Objections***

3. Claim 65 is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. Claim 65 refers to subsequent claim 66 instead of properly referring back to a previous claim.

4. Claim 43 is objected to because of the following informalities: the period (.) at the end of line 3 should presumably be a semicolon (;). Appropriate correction is required.

***Claim Rejections - 35 USC § 102***

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 38-41, 44-63, and 66-72 are rejected under 35 U.S.C. 102(b) as being anticipate by Binoy Ravindran and Lonnie R. Welch, “Modeling and Analysis of Complex, Dynamic Real-time Systems,” 1998, The University of Texas at Arlington, AAT 9904954, pp. i-xviii, 1-241 (hereinafter “[RW1998]”).

As per claim 38, [RW1998] discloses:

providing instrumentation information to a resource allocation function (see, e.g., pp. 77-79 describing hardware monitors and software monitors that dynamically measure QoS metrics; p. 109 QoS information allows for allocation analysis), the instrumentation information being associated with N-plurality of hosts (p. 78—associated with the hardware monitors are host-level daemons (one per host machine) that collect various host-level metrics);

preparing system specification files to describe system and network specification information (see, e.g., p. 77—a parser that is front-end to the system data broker reads a description of the system and its QoS requirements expressed using a specification language; see also, chapter 7 on pp. 87-108, describing in detail the specification language for modeling host and network systems);

linking the system specification files to characteristic applications (see p. 77—a parser reads the specification files containing QoS requirements to build the data structures that model the system; p. 109—system components are monitored for conformance to the specified QoS requirements; pp. 142-47—the resources for the applications are allocated on the basis of the monitored QoS information; thus, the specification files are “linked to”, i.e., associated with, the operation of (and allocation of resources for) the applications);

producing quality-of-service (QoS) information by the resource allocation function based the instrumentation information (see, e.g., section 8.3.1 on pp. 143-45, describing the monitoring of hardware load for each host; the hardware load information is used by the resource allocation algorithm to generate a “fitness” score, as described in section 8.3.2 on pp. 145-47), the QoS information being associated with the characteristic applications on the N-plurality of hosts (as stated above, the hardware load for the hosts is monitored; see, e.g., section 8.3.1 on pp. 143-45);

allocating assigned hosts for processing the characteristic applications as control orders by the resource allocation function based on the QoS information (see, e.g., enumerated item “(3)” on p. 81—computing a resource allocation: the resource manager determines candidate hosts . . .; see also section 8.3 on pp. 142-47, describing allocation analysis which includes a resource allocation algorithm for the selection of a “best” resource for a recovery action; Figure 8.6 on p. 148 further illustrates the resource allocation algorithm), the assigned hosts being among the N-plurality of hosts (the algorithm considers the “eligible hosts” obtained from the system specification; p. 146); and

compiling commands for the respective characteristic applications by the application control function to the assigned hosts based on the control orders and the QoS information (see,

e.g., enumerated item “(4)” on p. 81, continuing onto p. 82—the program control enacts the allocation decision by contacting the startup daemons (which start or stop processes based on the type of request from the program control) on the respective hosts; see also Figure 8.6 on p. 148, describing a resource allocation algorithm).

As per claim 39, [RW1998] further discloses preparing the specification file being performed in a language that provides syntax adapted to describe the system and network specification information (see chapter 7 on pp. 87-108, describing in detail the specification language for modeling host and network systems).

As per claim 40, [RW1998] further discloses producing the QoS information further comprising analyzing the instrumentation information from the N-plurality of hosts by a corresponding N-plurality of QoS managers (p. 78—associated with the hardware monitors are host-level daemons (one per host machine) that collect various host-level metrics); and producing the QoS information by the QoS managers (p. 78—low-level metrics are sent by the the daemons to the hardware broker where higher-level metrics are computed).

As per claim 41, [RW1998] further discloses providing the instrumentation further comprising reporting application events of the applications from the N-plurality of hosts (see, e.g., p. 78, describing the receipt of time-stamped event tags from programs); collecting the application events as compiled events by an instrumentation collector (e.g., p. 78, the time-stamped events are gathered and transformed into path-level QoS metrics); correlating the compiled events as application metrics for the instrumentation information (p. 78—again, the time-stamped events are transformed into path-level QoS metrics); and providing the

instrumentation information to the QoS managers (p. 78—the metrics are evaluated for QoS violations).

As per claim 44, [RW1998] further discloses copying a characteristic application of the M plurality of managed characteristic applications by the QoS managers as a copy for an additional host among the N-plurality of hosts (starting replicas of bottleneck programs for a path; see, e.g., p. 81 enumerated items “(3)” and “(4)”).

As per claim 45, [RW1998] further discloses copying the characteristic application comprising transmitting a scale request associated with the copy to the resource allocation function (see, e.g., enumerated item “(4)” on p. 81—the resource manager notifies the program control about reallocation decisions, including starting replicas of programs).

As per claim 46, [RW1998] further discloses analyzing the instrumentation information and the system specification files by a resource manager to assign priorities to the characteristic applications (p. 141—a list of unhealthy applications, sorted by the degree of slowdown experienced, is generated); and distributing the characteristic applications to the assigned hosts based on the priorities (p. 145 the resource allocation algorithm determines the best host on which to replicate, migrate, or restart applications from the sorted list of unhealthy applications).

As per claim 47, [RW1998] further discloses determining respective loads of the N-plurality of hosts by a hardware broker, the loads corresponding to computation requirements by the hosts and the network (see, e.g., section 8.3.1 on pp. 143-45); and providing the loads to the resource manager (see, e.g., section 8.3.2 on pp. 145-47).

As per claim 48, [RW1998] further discloses receiving respective operational statuses of the N-plurality of hosts to the hardware broker from history servers associated with the hosts

Art Unit: 2192

(see, e.g., section 8.3.2 on pp. 145-47; Figure 8.6 on p. 148); assigning fitness scores associated with the respective operation statuses by the hardware broker (*Id.*); and determining the loads based on the fitness scores by the hardware broker (*Id.*).

As per claim 49, [RW1998] further discloses creating an instruction by a program control based on the control orders and the QoS information, the instruction being associated with a characteristic application of the M plurality of managed characteristic applications (see, e.g., enumerated item “(4)” on p. 81, continuing onto p. 82—the program control enacts the allocation decision by contacting the startup daemons (which start or stop processes based on the type of request from the program control) on the respective hosts); compiling the instruction by the program control (*Id.*); and submitting the compiled instruction to the assigned hosts (*Id.*).

As per claim 50, [RW1998] further discloses creating a process startup order in response to a notification of QoS violation from the resource allocation function (see, e.g., the enumerated list on pp. 81-82, and in particular, see item “(4)”).

As per claim 51, [RW1998] further discloses creating a process shutdown order in response to a notification of QoS violation from the resource allocation function (see, e.g., the enumerated list on pp. 81-82, and in particular, see item “(4)”).

As per claim 52, [RW1998] discloses:

providing instrumentation information to an N-plurality of quality-of-service (QoS) managers of a resource allocation function (see, e.g., pp. 77-79 describing hardware monitors and software monitors that dynamically measure QoS metrics; p. 109 QoS information allows for allocation analysis), the instrumentation information being associated with N-plurality of hosts

(p. 78—associated with the hardware monitors are host-level daemons (one per host machine) that collect various host-level metrics), the QoS managers being associated with the N-plurality of hosts (*Id.*);

preparing system specification files to describe system and network specification information (see, e.g., p. 77—a parser that is front-end to the system data broker reads a description of the system and its QoS requirements expressed using a specification language; see also, chapter 7 on pp. 87-108, describing in detail the specification language for modeling host and network systems);

linking the system specification files to characteristic applications (see p. 77—a parser reads the specification files containing QoS requirements to build the data structures that model the system; p. 109—system components are monitored for conformance to the specified QoS requirements; pp. 142-47—the resources for the applications are allocated on the basis of the monitored QoS information; thus, the specification files are “linked to”, i.e., associated with, the operation of (and allocation of resources for) the applications);

producing QoS information by the QoS managers based on the instrumentation information (see, e.g., section 8.3.1 on pp. 143-45, describing the monitoring of hardware load for each host; the hardware load information is used by the resource allocation algorithm to generate a “fitness” score, as described in section 8.3.2 on pp. 145-47), the QoS information being associated with the characteristic applications on the N-plurality of hosts (as stated above, the hardware load for the hosts is monitored; see, e.g., section 8.3.1 on pp. 143-45);

analyzing the instrumentation information and the system specification files by a resource manager of the resource allocation function (see, e.g., section 8.3 on pp. 142-47, describing

Art Unit: 2192

allocation analysis which includes a resource allocation algorithm for the selection of a “best” resource for a recovery action; Figure 8.6 on p. 148 further illustrates the resource allocation algorithm);

allocating assigned hosts for processing the characteristic applications as control orders by the resource manager based on the QoS information (see, e.g., enumerated item “(3)” on p. 81—computing a resource allocation: the resource manager determines candidate hosts . . .; see also section 8.3 on pp. 142-47, describing allocation analysis which includes a resource allocation algorithm for the selection of a “best” resource for a recovery action; Figure 8.6 on p. 148 further illustrates the resource allocation algorithm), the assigned hosts being among the N-plurality of hosts (the algorithm considers the “eligible hosts” obtained from the system specification; p. 146); and

compiling commands for the respective characteristic applications by a program controller of the application control function to the assigned hosts based on the control orders and the QoS information (see, e.g., enumerated item “(4)” on p. 81, continuing onto p. 82—the program control enacts the allocation decision by contacting the startup daemons (which start or stop processes based on the type of request from the program control) on the respective hosts; see also Figure 8.6 on p. 148, describing a resource allocation algorithm).

As per claim 53, [RW1998] further discloses analyzing the instrumentation information and the system specification files by a resource manager to assign priorities to the characteristic applications (p. 141—a list of unhealthy applications, sorted by the degree of slowdown experienced, is generated); and distributing the characteristic applications to the assigned hosts

Art Unit: 2192

based on the priorities (p. 145 the resource allocation algorithm determines the best host on which to replicate, migrate, or restart applications from the sorted list of unhealthy applications).

As per claim 54, [RW1998] further discloses creating an instruction by a program control based on the control orders and the QoS information, the instruction being associated with a characteristic application of the M plurality of managed characteristic applications (see, e.g., enumerated item “(4)” on p. 81, continuing onto p. 82—the program control enacts the allocation decision by contacting the startup daemons (which start or stop processes based on the type of request from the program control) on the respective hosts); compiling the instruction by the program control (*Id.*); and submitting the compiled instruction to the assigned hosts (*Id.*).

As per claim 55, [RW1998] further discloses determining respective loads of the N-plurality of hosts by a hardware broker, the loads corresponding to computation requirements by the hosts and the network (see, e.g., section 8.3.1 on pp. 143-45); and providing the loads to the resource manager (see, e.g., section 8.3.2 on pp. 145-47).

As per claim 56, [RW1998] further discloses receiving respective operational statuses of the N-plurality of hosts to the hardware broker from history servers associated with the hosts (see, e.g., section 8.3.2 on pp. 145-47; Figure 8.6 on p. 148); assigning fitness scores associated with the respective operation statuses by the hardware broker (*Id.*); and determining the loads based on the fitness scores by the hardware broker (*Id.*).

As per claim 57, [RW1998] further discloses copying a characteristic application of the M plurality of managed characteristic applications by the QoS managers as a copy for an additional host among the N-plurality of hosts (starting replicas of bottleneck programs for a path; see, e.g., p. 81 enumerated items “(3)” and “(4)”).

As per claim 58, [RW1998] further discloses copying the characteristic application comprising transmitting a scale request associated with the copy to the resource allocation function (see, e.g., enumerated item “(4)” on p. 81—the resource manager notifies the program control about reallocation decisions, including starting replicas of programs).

As per claim 59, [RW1998] discloses:

a plurality of quality-of-service (QoS) managers corresponding to N-plurality of hosts (p. 78—associated with the hardware monitors are host-level daemons (one per host machine) that collect various host-level metrics), the QoS managers receiving instrumentation information from respective hosts and producing QoS information based on the instrumentation information (see, e.g., section 8.3.1 on pp. 143-45, describing the monitoring of hardware load for each host; the hardware load information is used by the resource allocation algorithm to generate a “fitness” score, as described in section 8.3.2 on pp. 145-47), the instrumentation information being associated with the N-plurality of hosts (p. 78—the hardware monitors/host-level daemons collect various host-level metrics), the QoS information being associated with characteristic applications on the N-plurality of hosts (as stated above, the hardware load for the hosts is monitored; see, e.g., section 8.3.1 on pp. 143-45);

a library (a collection of model data structures) that links system specification files that describe system and network specification information, the library linking the specification files to the characteristic applications (see p. 77—a parser reads the specification files containing QoS requirements to build the data structures that model the system; p. 109—system components are monitored for conformance to the specified QoS requirements; pp. 142-47—the resources for the

applications are allocated on the basis of the monitored QoS information; thus, the collection of model data structures derived from the specification files are “linked to”, i.e., associated with, the operation of (and allocation of resources for) the applications);

a resource manager that allocates assigned hosts for processing the characteristic applications as control orders based on the QoS information (see, e.g., enumerated item “(3)” on p. 81—computing a resource allocation: the resource manager determines candidate hosts . . . ; see also section 8.3 on pp. 142-47, describing allocation analysis which includes a resource allocation algorithm for the selection of a “best” resource for a recovery action; Figure 8.6 on p. 148 further illustrates the resource allocation algorithm), the assigned hosts being among the N-plurality of hosts (the algorithm considers the “eligible hosts” obtained from the system specification; p. 146); and

a program controller that compiles commands for the respective characteristic applications to the assigned hosts based on the control orders and the QoS information (see, e.g., enumerated item “(4)” on p. 81, continuing onto p. 82—the program control enacts the allocation decision by contacting the startup daemons (which start or stop processes based on the type of request from the program control) on the respective hosts; see also Figure 8.6 on p. 148, describing a resource allocation algorithm).

As per claim 60, [RW1998] further discloses: an instrumentation collector that receives application events as compiled events (e.g., p. 78, the time-stamped events are gathered and transformed into path-level QoS metrics), the application events being reported from the N-plurality of hosts (see, e.g., p. 78, describing the receipt of time-stamped event tags from

programs); an instrumentation correlator that correlates the compiled events as application metrics for the instrumentation information (p. 78—again, the time-stamped events are transformed into path-level QoS metrics).

As per claim 61, [RW1998] further discloses a hardware broker that analyzes history server information and provides respective loads of the N-plurality of hosts to the resource manager (see, e.g., section 8.3.1 on pp. 143-45; section 8.3.2 on pp. 145-47), the loads corresponding to computation requirements by the hosts and the network (see, e.g., section 8.3.1 on pp. 143-45).

As per claim 62, [RW1998] further discloses history servers providing operation statuses of the hosts to the hardware broker determining the respective loads (see, e.g., section 8.3.2 on pp. 145-47; Figure 8.6 on p. 148).

As per claim 63, [RW1998] further discloses the loads being determined from fitness scores, and the fitness scores being determined from the operational statuses (see, e.g., section 8.3.2 on pp. 145-47; Figure 8.6 on p. 148).

As per claim 66, [RW1998] further discloses the program controller communicating with the hosts using operating systems (see, e.g., p. 100, describing the machine-specific startup and shutdown characteristics including the name and version of the operating system, and the locations of executable scripts).

As per claim 67, [RW1998] further discloses at least one of the up to M managed characteristic applications comprising a scalable application (see, e.g., pp. 99-100, describing the definition of an application as including scalability properties; see also section 7.1 on pp. 101-03, discussing scalability in more detail).

As per claim 68, [RW1998] further discloses at least one of the up to M managed characteristic applications comprising a fault tolerant application, where the degree of fault tolerance is user selectable (see, e.g., section 7.1 on pp. 101-03, describing a required redundancy level specified as an integer).

As per claim 69, [RW1998] further discloses the instrumentation information providing condition metrics of the characteristic applications, the conditions including at least one of execution status, computation cycles, running duration and memory usage (see, e.g., the description of the hardware monitors and software monitors on pp. 78-79).

As per claim 70, [RW1998] further discloses an N-plurality of program control displays corresponding to the hosts and communicating with the program controller (see, e.g., the description of the human computer interface (HCI) on pp. 79-80), the displays displaying information from the resource manager (*Id.*).

As per claim 71, [RW1998] further discloses the QoS managers copying a characteristic application of the M plurality of managed characteristic applications as a copy for an additional host among the N-plurality of hosts (starting replicas of bottleneck programs for a path; see, e.g., p. 81 enumerated items “(3)” and “(4)”).

As per claim 72, [RW1998] further discloses the QoS managers transmitting a scale request associated with the copy to the resource manager (see, e.g., enumerated item “(4)” on p. 81—the resource manager notifies the program control about reallocation decisions, including starting replicas of programs).

***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 42 and 64 are rejected under 35 U.S.C. 103(a) as being unpatentable over [RW1998] in view of Tony DeWitt, et al., "ReMoS: A Resource Monitoring System for Network-Aware Applications," Carnegie Mellon University, Tech. Report CMU-CS-97-194, 1997, pp. 1-30 (hereinafter "[DeW1997]").

Regarding claims 42 and 64, [RW1998] discloses all of the features recited in parent claims 40 and 64 (see the § 102(b) rejections above) but fails to expressly disclose providing an application programming interface (API) that enables the QoS managers to access the specification information using API calls. However, [DeW1997] teaches the use of such an API in the context of a resource monitoring system that collects static and dynamic information to allow network-aware applications to tune their execution behavior to the dynamic state of the network, where API calls are used to access the specification information (e.g., [DeW1997] Abstract, providing a general overview; [DeW1997] §§ 5 and 6, describing the API and its usage; [DeW1997] Appendix A, providing the details of the API structures and functions). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to combine the use of such an API with the system disclosed by [RW1998]. One would be motivated to do so to gain the advantages of providing a uniform interface so that,

Art Unit: 2192

for example, portable network-aware applications can be developed independent of any particular network architecture (e.g., [DeW1997] § 9).

***Allowable Subject Matter***

9. Claims 43 and 65 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The following is a statement of reasons for the indication of allowable subject matter: The prior art of record fails to teach or fairly suggest specifically providing C++ classes for storing specification file information in the context of the API of claims 43 and 65.

***Conclusion***

10. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Art Unit: 2192

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry of a general nature should be directed to the TC 2100 Group receptionist: 571-272-2100.

A handwritten signature in black ink, appearing to read "Eric B. Kiss". The signature is stylized with a large "E" and "K".

Eric B. Kiss  
June 9, 2006